## Git Cheat Sheet Summary

**Stage & Snapshot**

* **git status**: Shows modified files in the working directory, staged for your next commit.
* **git add [file]**: Adds a file as it currently appears to your next commit (staging area).
* **git reset [file]**: Unstages a file while keeping the changes in the working directory.
* **git diff**: Shows differences between the working directory and the staging area.
* **git diff --staged**: Shows differences between the staging area and the last commit.
* **git commit -m "[descriptive message]"**: Commits the staged content as a new commit snapshot.

**SETUP & INIT**

* **git config --global user.name "[firstname lastname]"**: Sets your name for version history.
* **git config --global user.email "[valid-email]"**: Sets your email address for version history.
* **git config --global color.ui auto**: Enables automatic command line coloring for Git.
* **git init**: Initializes an existing directory as a Git repository.
* **git clone [url]**: Retrieves an entire repository from a remote location via URL.

**BRANCH & MERGE**

* **git branch**: Lists all branches, with an asterisk next to the active one.
* **git branch [branch-name]**: Creates a new branch at the current commit.
* **git checkout [branch]**: Switches to another branch and checks it out into your working directory.
* **git merge [branch]**: Merges the specified branch’s history into the current branch.
* **git log**: Shows all commits in the current branch’s history.

**SHARE & UPDATE**

* **git remote add [alias] [url]**: Adds a Git URL as an alias.
* **git fetch [alias]**: Fetches all branches from a remote repository.
* **git merge [alias]/[branch]**: Merges a remote branch into the current branch.
* **git push [alias] [branch]**: Transmits local branch commits to the remote repository.
* **git pull**: Fetches and merges any commits from the tracking remote branch.

**TRACKING PATH CHANGES**

* **git rm [file]**: Deletes a file from the project and stages the removal for commit.
* **git mv [existing-path] [new-path]**: Changes an existing file path and stages the move.
* **git log --stat -M**: Shows all commit logs with indication of any paths that moved.

**TEMPORARY COMMITS**

* **git stash**: Saves modified and staged changes.
* **git stash list**: Lists the stack order of stashed file changes.
* **git stash pop**: Applies the changes from the top of the stash stack.
* **git stash drop**: Discards the changes from the top of the stash stack.

**REWRITE HISTORY**

* **git rebase [branch]**: Applies any commits of the current branch ahead of the specified branch.
* **git reset --hard [commit]**: Clears the staging area and rewrites the working tree from the specified commit.

**INSPECT & COMPARE**

* **git log**: Shows the commit history for the currently active branch.
* **git log branchB..branchA**: Shows commits on branch A that are not on branch B.
* **git log --follow [file]**: Shows commits that changed a file, even across renames.
* **git diff branchB...branchA**: Shows the difference between branch A and branch B.
* **git show [SHA]**: Shows any Git object in human-readable format.

**IGNORING PATTERNS**

* **git config --global core.excludesfile [file]**: Sets a system-wide ignore pattern for all local repositories.
* **.gitignore file**: Contains patterns for ignoring files and directories.